Pipeline Leak Detection System in a Palm Oil Fractionation Plant Using Artificial Neural Network

Arshad Ahmad[1] **Mohd. Kamaruddin Abd. Hamid**[2]

[1]*Department of Chemical Engineering*
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
Tel: +60-7-553-5502, Fax: +60-7-558-1463, Email: arshad@fkkksa.utm.my

[2]*Laboratory of Process Control, Department of Chemical Engineering*
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia
Tel: +60-7-553-5858, Email: kamaruddin@rmc.utm.my

Abstract

A leak detection system for pipelines is designed and tested. Detection of leak in pipelines is an important task for economical and safety operation, loss prevention and environmental protection. Therefore, a leak detection of pipelines plays an important role in the plant safety operation. In this paper, a neural network based detection scheme integrating a neural Elman network dynamic predictor and a feedforward neural network fault classifier is proposed to overcome the problem of leak detection. The scheme was implemented to detect leakage in a palm oil fractionation process. To generate the required simulation data, Hysys.Plant dynamic process simulator was employed. The use of the output prediction error, between a neural network model and a non-linear dynamic process, as a residual for detecting leakage faults is analysed. A second neural network classifier is developed to detect the leak from the residuals generated, and results are presented to demonstrate the satisfactory detection of leakage achieved using this scheme that can detect leak as small as 0.1%.

Keywords:

Leak detection; neural networks; palm oil fractionation process

Introduction

Pipelines in a palm oil fractionation plant are used to transport fluids from one component to another. Leaks in pipelines carrying fluids or solvents can cause serious pollution, injuries and fatalities, if they are not promptly detected and repaired. Large leaks cause significant changes in pressure gradients and differences in mass flow rates at measurement points, and therefore are easy to detect. On the other hand, small leaks are more difficult to detect because changes in the usual process measurements are small. However, leaks as small as 1% of the nominal flow rate can cause the discharge of a large amount of dangerous fluid before they are detected, usually by the impact they have on the surrounding environment. The early detection of such small leaks is then the main goal of a leak-detection system.

Many methods for creating leak-detection systems in liquid and gas pipelines have been proposed, mainly based on process variables (pressure, flow rate, and temperature). Usually measured in pipelines. Stouffs and Giot [1] present some mass-balance-based systems, using a pipeline flow model in order to compute changes in the pipeline inventory during transient flow. They highlight the importance of the packing term, and conclude that the bottom line for leak detection would be approximately 2% for steady-state flow and 3% for transient flow.

Billmann and Isermann [2] proposed a method based on a nonlinear adaptive way of observing the pipeline dynamics and a special correlation technique for fault detection based on flow and pressure measurements at pipeline inlet and outlet. Nonlinear pipeline models have been used, and the friction coefficient estimated on-line. This leads to an adaptive (nonlinear) observation method. Billmann and Isermann show that the detectable leaks were greater that 2% for liquid and 10% for gas.

Hamande et al. [3] discuss a model-based leak-detection system that was installed on an ethylene pipeline. The system has been in operation since 1989, and is regularly tested by valving product through flares along the pipeline. Alarms are generated when the mass imbalance becomes larger than a given threshold. Measurements of pressures and temperatures are performed at 21 locations along the pipeline, including inlet and outlet, where the mass flow rate is also measured. The system uses a real-time, transient flow model - fed by pressure and temperature measurements - that simulates the detailed design of the pipeline. The smallest detectable leak is about 7% in 1 hour (1 ton/h, the nominal flow rate in the pipeline being 15 ton/h).

Zhang [4] describes a statistical method for measuring and locating leaks in pipelines. The system uses flow and pressure measurements at the ends of a pipeline, and has been applied to detect leaks in a simulated 100-km-long gas pipeline. It has also been field tested on a 37-km-long propylene pipeline. Both the numerical simulation and the field test showed that the scheme can detect leaks as small as 1% of the nominal flow rate as well as pinpoint their locations with high accuracy.

Finally, pressure waves generated by the leak provide another potential method of leak detection by measuring the pressure disturbances that travel along the line [5]. Signal conditioning is required in order to monitor the temperature and pressure variation in the pipeline (correcting the velocity of the sound for any variation) and to account for process operations, eliminating the pressure disturbances deriving from normal processes.

Artificial neural networks (ANN) have attributes that can actually make them good for processing routine measurements made in pipelines, and can be used in quite innovative leak-detection systems, without requiring very high sampling frequencies. An ANN can be regarded as a nonlinear mathematical function that transforms a set of input variables into a set of output variables [6], the transformation function depending on weights that are determined on the basis of a set of training examples. Once weights have been calculated, processing of new data is fast. In addition to offering very high processing speed, ANNs are, in principles capable of learning a general solution to a problem from a limited number of examples. Be that as it may, the use of ANNs does not appear to have received much attention for leak detection, as yet. However, there are many successful applications in tasks of similar complexity. Some of the applications can be reviewed in [7-8]. There are some specific reasons why using ANNs for leak detection is quite promising [6], namely, (1) it is difficult to find an adequate first-principle or model-based solution; (2) new data must be processed at high speed, and (3) the system must be impervious to noise. Another important aspect for the development of ANNs is that a large set of data must be available for training purposes. In general, such information is either available or can be developed for pipelines.

The main goal of this work was to develop a leak-detection system capable of detecting leaks down to 1% or less of the nominal flow rate at an acceptable cost. The leak-detection system is based on process variables routinely measured during operations, and ANNs are utilized in order to process the field data. In the specific case of pipelines in the fractionation plant, the data for ANN development should consist of field measurements of process variables (pressure, flow rate, temperature) performed where there are leaks. Cases where there are no leaks can also be used for part of the training operation, but, ideally, data measured when there are actual leaks should be available. This paper elaborates the application of neural networks in leaks detection. In the following sections, an introduction to ANN will be presented. This will be followed by the case studies with results, discussions and conclusion drawn from the work.

Neural Network Leaks Detection Scheme

The proposed leaks detection scheme is hierarchical in structure. The schematic diagram of the strategy is shown in Figure 1 below. In short, there are two types of model required within this scheme. Both are developed using neural network. The first is the estimator that will always estimate the "normal" or fault-free process behaviour. Next, there is the leak detector that will identify the sources of leak that takes place.

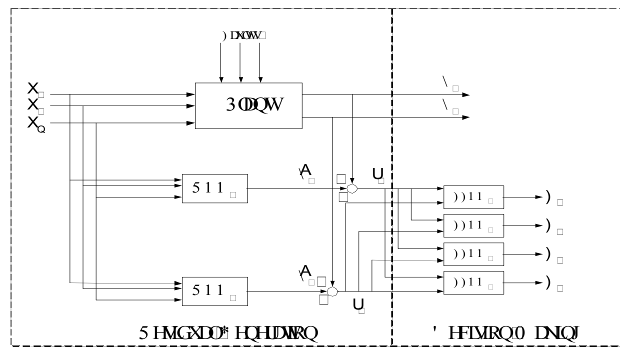

Figure 1 – The Neural Network-Based Leaks Detection Scheme

The hierarchical approach is advantageous because it alleviates the chances of misidentification of normal operation trend that is due to the manipulation of the feeds condition. In practice, there are always possibilities that the manipulation of feeds will produce process conditions that coincidentally match the leaks pattern and the detector will tend to misinterpret the situation. The use of residuals provides some protection to the system.

Artificial Neural Network

Artificial Neural Network (ANN) is a computing tool that is inspired by the capabilities of human brains. An ANN is constructed of interconnected basic elements called nodes or neurons. A schematic diagram of this neuron is shown in Figure 2. Similar to their biological counterpart, these neurons are capable of processing incoming information and transferring them to other neurons.

The input signals come from either the environment or outputs of other neurons through connections as specified by the network architecture. Within each neuron, input signals are summed and transformed using a specified activation function before being sent to other neurons. Such transformations are needed to impart pattern-mapping capability to the networks.

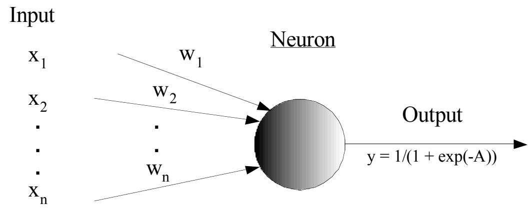

Figure 2 - An Individual Neural Network Processing Unit (Neuron).

A commonly used function known as the sigmoid is given by the following equation:

$$f(z) = (1 + e^{-z})^{-1} \qquad (1)$$

Here, z is a weighted sum of all inputs. Associated with each connection is an adjustable value called network weights. During learning, these weights are adjusted to fulfil the training objective. Effectively, network weights serve as a measure for connection strength that controls the influence of each incoming signal on the recipient neuron.

Feedforward Network

In process engineering applications, the most commonly used network architecture is the multilayer feedforward network as displayed in Figure 3. This network is also known as multilayer perceptron. It is constructed of neurons arranged in several layers. There is an input layer to receive the incoming data to the network, and an output layer to deliver the processed data from the network. In between these two layers, there could be several layers known as the *hidden layers*. Except for those in the input layer, all neurons carry out information processing as mentioned earlier.

The selection of input variables is carried out using various considerations. The aim is to include all strong relationships and neglecting some of the weaker links for the sake of model simplicity. Experiences have shown that by feeding some of the delayed values of the outputs as inputs to the network, the prediction capabilities are improved significantly.

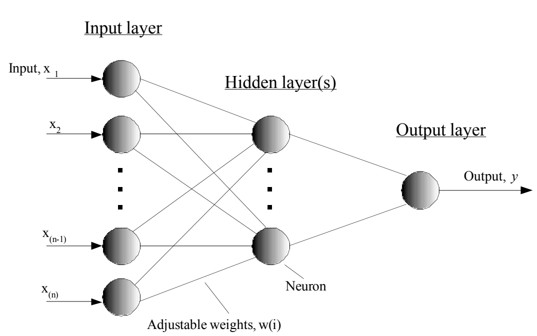

Figure 3 - A Three-layer Feedforward Neural Network.

The input-output relationships between the variables can be represented by equation (2) below:

$$\hat{y}_i = f\{u_1(k),\ldots u_1(k-n); u_2(k),\ldots u_2(k-n); \ldots; y_i(k-1)\ldots y_i(k-m)\} \qquad (2)$$

where u_i is the input and $\hat{y}_i$ is the predicted output and $y_i(k-1)$ is the delayed output signals. The number of delayed signals to be used for the case of both input and output variables depends on the process. Looking from the perspective of linear modelling, decisions can be based on the model order. However, this may not be strictly followed here. The main aim is to accommodate the effect of time delays and any uncertainties associated with them. In this study, feedforward network architecture has been used for fault classifier.

For the development of the estimator to be implemented in the proposed scheme, feedforward network with delayed signals of the output fed as inputs cannot be employed. This is because the outcome of such models will be the actual operating conditions. This means that on occasion of leak conditions, the model will not be able to estimate the desired normal, leak-free conditions. For such applications, the use of recurrent networks (RNN) is favourable. RNN involving dynamic elements and internal feedback connections have also been suggested to be superior for modelling non-linear systems compared to feedforward networks [9]. Various works have been presented showing that recurrent neural networks are quite effective in modelling non-linear dynamical systems [10-11]. Elman network falls into this category.

Elman Network

Elman [12] has proposed a partially recurrent network, where the feedforward connections are modifiable and the recurrent connections are fixed. Figure 4 shows the schematic diagram of this network architecture.

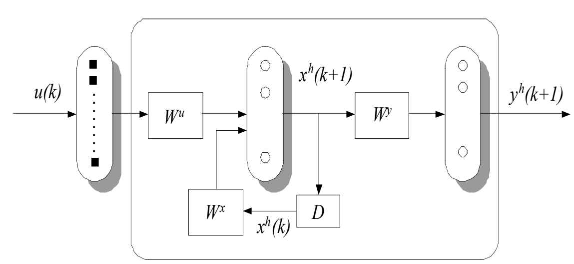

Figure 4 – Block Diagram of Elman Network

To understand the feature offered by Elman net, consider a multivariable plant with m inputs and q outputs, describe by a general non-linear input-output discrete time state space model:

$$x(k+1) = f\{x(k), u(k)\} \qquad (3)$$

$$y(k) = g\{x(k)\} \qquad (4)$$

where $f:\Re^{n+p} \rightarrow \Re^{n}$ and $g:\Re^{n} \rightarrow \Re^{q}$ are non-linear functions; $u(k) =\in \Re^{m}$, $y(k) \in \Re^{q}$ and $x(k) \in \Re^{n}$ are, respectively, the input vector, the output vector and the state vector, at a discrete time k. In addition to the input and the output units, the Elman network has a hidden unit, $x^{h}(k) \in \Re^{n}$. $W^{x} \in \Re^{nxn}$, $W^{u} \in \Re^{nxp}$ and $W^{y} \in \Re^{qxn}$ are the interconnection matrices, respectively, for the context-hidden layer, input-hidden layer and hidden-output layer. Theoretically, an Elman network with n hidden units is able to represent an n^{th} order dynamic system.

The dynamics of the Elman network is describe by the difference equations (5) – (7).

$$s(k+1) = W^{x} x^{h}(k) + W^{u} u(k) \tag{5}$$

$$x^{h}(k+1) = \varphi\{s(k+1)\} \tag{6}$$

$$y^{h}(k+1) = W^{y} x^{h}(k+1) \tag{7}$$

where $s(k) \in \Re^{n}$ is an intermediate variable and $\varphi(.)$ is an hyperbolic tangent function.

Network Training

An important part of the neural network model development is the training stage where the optimum connection weights are determined through some selected optimisation algorithm. During training, the optimisation algorithm (often referred to as learning rule) adjusts the network weights so that the error between the actual output and the target output is minimized. One commonly used error criterion is the mean squares error given by equation (8) below:

$$\mathrm{E} = \frac{1}{N} \sum_{\mathrm{i=1}}^{\mathrm{N}} (\mathrm{t}_{\mathrm{i}}^{(\mathrm{m})} - \mathrm{y}_{\mathrm{i}}^{(\mathrm{m})})^{2} \tag{8}$$

Here, N denotes the number of training data presented to the input layer, $t_i^{(m)}$ represents the desired value of the ith output element given the mth data, while $y_i^{(m)}$ is the actual output of the same element.

Selection of Network Topology

Another important stage in neural network model development is the selection of network topology, i.e., the number of neurons in each of the network layer and its connection. Once the type of the network has been chosen, e.g., multilayer feedforward network, the orientation of network connection is defined. Similarly, the number of neurons in the output and input layers are also defined by the problem and hence, the emphasis on the topology selection is to determine the number of neurons in the hidden layer. The aim is to develop models that are accurate and robust. Neural network is known to be able to map any continuous function to any arbitrary accuracy [13]. This implies that the network can learn the relationship between any set of inputs and outputs so that when given the inputs, the outputs can be reproduced. To obtain sufficiently good approximation qualities, a network with "sufficient" neurons must be trained. In doing so, the weights associated with all the connections within the network are optimised to achieve the desired input/output mapping.

In developing the model, what is important is the ability of the model to predict the behaviour on “unseen” process. Therefore, the decision on the “optimal” model structure should not be made based on the performance of the model to reproduce the training set only because the resulting model may not fulfil the robustness requirement. Although there are many methods or criteria that are available to determine the best fit of the model parameters, in this work cross validation approach is adopted.

The concept of topology selection using cross-validation is that after estimation using a given sample of data, the quality of the mapping is evaluated using a different set of data. The best mapping is defined as the one that minimises the prediction error on a data set for which it was not trained. This approach of topology selection involves iterative efforts. Beginning with some small number of hidden neurons, the search continues until the desired performance is achieved.

Process Description

The application of neural network in leaks detection is implemented to a palm oil fractionation plant. The process consists of five distillation columns connected in series. A simplified schematic diagram of the plant is provided in Figure 5. The feed to the plant is Palm Kernel Oil (PKO), which contains fatty acids from C6 - C18.

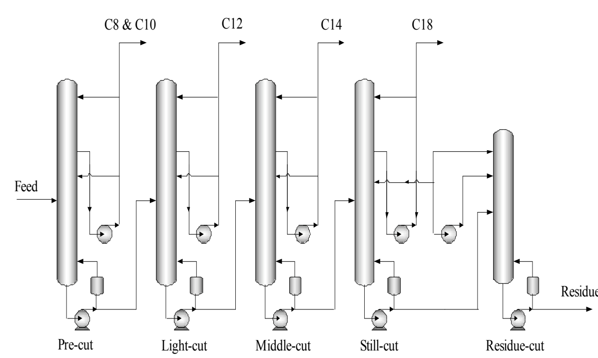

Figure 5 – Schematic Diagram of a Fractionation Process

In the pre-cut column, light products consisting of mainly C8 and C10 and a trace of C6 are recovered in the overhead. The bottom stream is fed to the light-cut column where C12 is separated from the rest of its constituents. The bottom product then enters the middle-cut column where C14 and C16 are recovered leaving the C18 to be purified from the rest of the oil constituents in the still-cut column. In the residue cut column, C18 is recovered and recycled back to the previous column. All the distillation columns operate at highly vacuum conditions that are generated using steam ejectors. The columns are packed with structured packing to provide the desired separation properties.

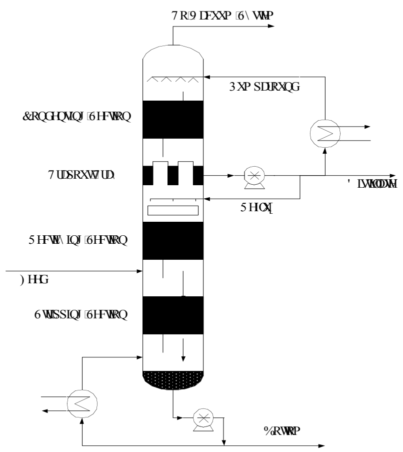

Figure 6 – A Typical Packed Column

Due to high operating temperatures, thermal oil is used for heating in the reboilers. A sectional view of a typical column is shown in Figure 6. The column is equipped with a pump-around system. Liquid collected on the trap-out tray is drawn out from a side-draw stream. The stream is split into two – a reflux stream and a stream that goes through an external cooler. The cooled stream is split again into two streams, one returning to the column as a recycle and one as the distillate. The pump-around system provides a means for the vapour in the column to be condensed through direct contact with the cooled liquid from above.

Results and Discussions

This investigation focused on the pipeline leak detection in the pre-cut column. Faulty conditions are simulated using the HYSYS.Plant software. Leaks are created causing the normal process operation to shift to a faulty operation mode. Effects of these leaks are expressed by the composition of C8 in the overhead stream leaving the pre-cut column.

Simulation of the column was carried out using HYSYS.Plant. Due to the non-conventional nature of the palm oil distillation system, the development of the flowsheet for the simulation has not been straightforward. For example, one of the long-chained fatty acid, caprylic acid (C8) is not found in HYSYS Component Properties Library. As such, C8's properties have to be estimated by HYSYS Hypothetical Component Manager. One has to provide as much data as possible so the estimation will be realistic. HYSYS.Plant does not support packed column as such, the equivalent tray (HETP) calculation was used. Similarly, model of a direct contact internal condenser based on packed column is also not available within the standard library and modifications have also been implemented. Proper initial values should be chosen for these streams; otherwise the system might converge to different values, which is not desirable due to the non-linearity and unstable characteristics of the process [14].

In this study, the ANN model development efforts were carried out using the neural network toolbox available within MATLAB software.

Process Estimator

A number of important issues must be addressed when dealing with the development of the neural network estimator. The first step was selecting the input variables and output variable for the neural network estimator. The inputs were selected based upon their availability in the actual industrial column and their effects on the top product composition. The inputs have been identified to consist 12 variables; column top temperature, column bottom temperature, column middle temperature, column top pressure, reflux flowrate and distillate flowrate as well as a two-sampling-interval-delayed signals of each these variables. The output variable is the composition of C8 (% mass) in the distillate stream.

Plant test data generation was designed and conducted in Hysys.Plant with emphasis to capture the dynamic behaviour of the column. A data set consisting of 1500 data points was obtained. The data was divided into three sets, a training set, a validation set, and a testing set.

The next stage concerned with the development of an artificial neural network (ANN) model that correctly mapped the input variables to the output variable. Elman network with structure shown in Figure 4 had been chosen to train the data. The Elman network has *tansig* (hyperbolic tangent sigmoid transfer function) neurons in its hidden (recurrent) layer, and *purelin* (linear transfer function) neurons in its output layer. With this combination, the network can approximate any function even with a finite number of discontinuities to an arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fit increases in complexity.

Network training was implemented using Levenberg-Marquardt Backpropagation (trainlm). The network was cross-validated at every batch training and thus the cross-validation errors of the network were monitored throughout the training. Network weights and biases were selected based on the minimum cross-validation error achieved in the training.

Table 1 - Training Result of MISO networks with 2 delayed terms and Trainlm

No. of Hidden Nodes	C8 FLOWRATE	
	Training Error	Validation Error
5	5.630093e-3	6.850242e-3
6	2.176698e-3	4.759817e-3
7	6.022385e-4	6.841415e-3
8	5.118487e-3	1.419495e-2
9	5.088225e-4	5.527025e-3
10	1.369624e-3	4.302835e-3
11	2.721438e-3	2.791515e-3
12	2.707649e-3	7.579502e-3
13	2.668948e-3	3.587602e-3
14	1.267853e-3	5.266927e-3
15	2.458897e-3	4.459251e-3
16	5.001075e-3	2.526958e-2
17	1.647529e-3	6.635936e-3
18	3.439583e-3	1.513190e-2
19	1.367297e-3	6.753807e-3
20	9.460975e-4	2.664814e-2
21	4.500387e-3	7.815148e-3
22	5.695127e-3	5.501059e-2
23	1.733385e-3	2.497227e-2
24	2.296181e-2	1.991131e-2
25	1.335754e-4	1.216002e-2

Figure 7 - Training and Validation of Neural Network Model

The results revealed that the optimum network structure was established with 11 hidden neurons. Figure 7 displays the performance of ANN in tracking the actual process data during the training and validation stages. Good performance in the validation set indicated that the network was able to represent the behaviour of the process in different operating conditions than that of the training set. In other words, the network is capable of estimating the product composition based on "unseen" data.

Leak Detector

The leak detector was constructed using multi layer feedforward network with single hidden layer. Network training was implemented using Levenberg-Marquardt learning algorithm. The network was also cross-validated at every batch of the training.

For leakage fault, leak is simulated by creating a two output Tee-junction in the To Condenser pipeline. For the leak line, the percentage of leakage is controlled by a control valve as shown in Figure 8.

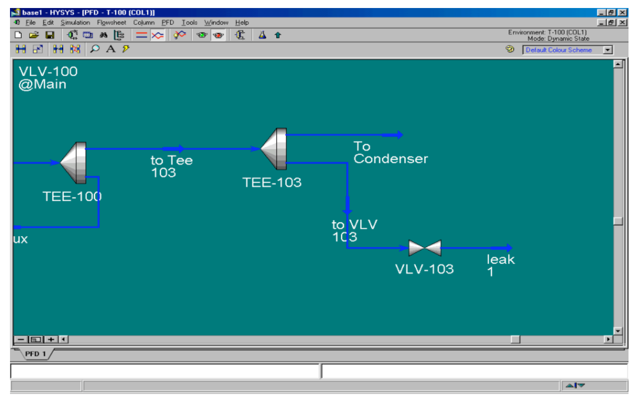

Figure 8 - Leakage Simulation in HYSYS Simulator

Leaks were simulated as if there were sudden failures of the measurement systems resulting from the leak. The leaks influence not only the performance of the control loops within the plant, but also the stability of the process. For example, when 0.3% and 0.5% leaks were introduced to the To Condenser pipeline (F5), the residual of C8 flowrate exceed its limit. Some of these impacts are shown in Figure 9 below.

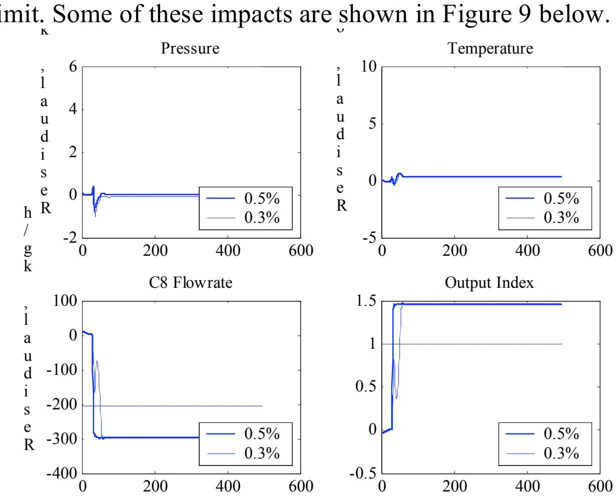

Figure 9 – Effect of Leak to the Distillate C8 Flowrate

The output data for leaks were designed to spread linearly between 0 and 1 with the index 0 and 1 being used to represent the process during normal condition and violation of process limit respectively. The index of 1 was used to indicate a process fault.

The network employed the C8 flowrate as an input. The outputs were the various leaks F1, F2, F3, F4, and F5. Training was accomplished using Levenberg-Marquadt algorithm and a network with 5 nodes in the hidden layer was found optimum.

Table 2 - Training results of MISO networks leak detector

No. of Hidden Nodes	TO CONDENSER PIPELINE LEAKAGE (F5)	
	Training Error	Validation Error
5	2.353716e-5	1.109019e-5
6	3.839039e-5	1.024365e-3
7	8.834487e-5	3.527061e-3
8	4.875869e-5	1.178918e-4
9	4.784626e-3	4.214639e-3
10	5.114614e-5	2.771619e-3
11	1.366973e-4	5.765655e-3
12	3.005566e-4	3.676785e-4
13	1.312997e-4	7.731330e-5
14	8.323252e-3	2.466752e-3
15	1.227990e-4	2.201055e-4
16	8.468933e-5	2.129597e-3
17	4.240486e-6	2.444376e-4
18	1.207543e-6	6.352933e-4
19	8.289171e-4	1.399145e-3
20	7.987803e-3	1.331796e-3
21	7.206751e-5	1.715392e-4
22	1.094338e-5	4.752952e-3
23	1.143766e-3	2.282391e-4
24	1.520425e-4	7.900378e-3
25	5.824322e-4	1.190514e-3

Leakage in the To Condenser stream was simulated by creating a Tee-junction with two outputs in that stream. Leakage is simulated by opening the control valve at the one of the two outputs. By opening the valve with certain percentage, we assume that the real leakage happened to the stream. Descriptions of the simulated leakage are given in the Table 3. The leak patterns and outputs of each detector are depicted in following figures.

Table 3 - Simulation of leakage for leak classifier performance testing

Fault Pattern	Leakage	Fault	Simulated Condition	Violation of Operating Limit
1	To Condenser stream	F5 (0.1%)	0.1% valve opening	Yes
2	To Condenser stream	F5 (0.3%)	0.3% valve opening	Yes

The results obtained revealed the success of the detector in detecting the leakage introduced to the system. For example, Figure 10 illustrates the detection of leak To Condenser stream (F5). When a 0.1% leak introduced to stream, the detector F5 successfully detected the leakage. This is clearly displayed in Figure 10 where the network output exceeded index of 1, as designed to indicate a process leakage.

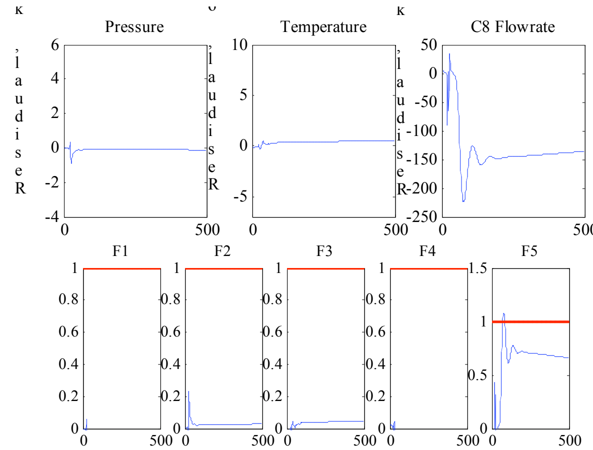

Figure 10 - F5 (0.1%) happened after 0.1% valve opening

Similarly, when 0.3% leak introduced to the stream, the outputs of detector F5 exceeded the index 1 to acknowledge the leakage condition as shown in Figure 11. Again, the process leakage was successfully detected.

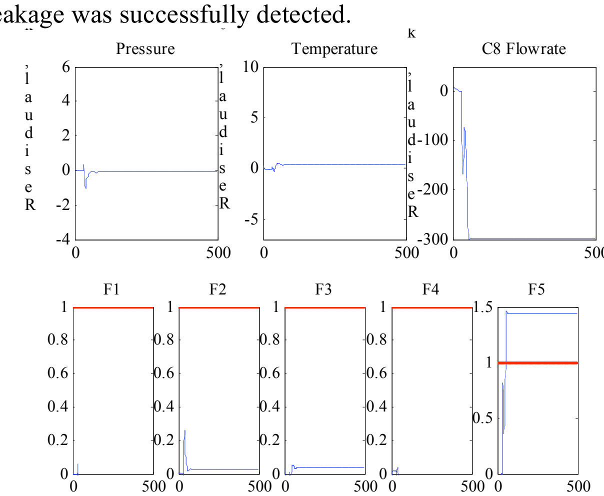

Figure 11 - F5 (0.3%) happened after 0.3% valve opening

Conclusion

An ANN-based system for leak detection has been developed. The leak-detection system consists of two stages. The first stage is the process estimator that will always estimate the process behaviour. The difference between the actual data with estimated data called residual is used as an input for the stage. The second stage is the detection system. In this stage, residual from the first stage will be evaluated. The leak-detection neural networks were trained and successfully tested leakage data. The system was able to detect as small as 0.1% of the C8 flow rate.

References

[1] Stouffs, P., and M. Giot, 1993. Pipeline Leak Detection Based on Mass Balance: Importance of the Packing Term, *J. Loss Prev. Process Ind.,* **6**, 5.

[2] Billmann, L., and R. Isermann, 1987. Leak Detection Methods for Pipelines, *Automatica,* **23**, 3.

[3] Hamande, A., V. Condacse, and J. Modisette, 1995. New System Pinpoints Leaks in Ethylene Pipeline, *Pipeline Gas Journal*.

[4] Zhang, X. J., 1992. Statistical Method for Detection and Localization of Leaks in Pipelines, *1992 OMAE, Vol. V-B, Pipeline Technology*, ASME, New York.

[5] Silk, M. G., and P. Carter, 1995. A Review of Means of Pipeline Leak Detection Independent of Flow Measurement, *EC Project DEPIRE DGXII Meeting*, Harwell, UK.

[6] Bishop, C. M., 1994. Neural Networks and Their Application, *Rev. Sci. Instrum.,* **65**, 6.

[7] Ahmad, A., and Abd. Hamid, M. K., 2002. Detection of Sensor Failure in a Palm Oil Fractionation Plant Using Artificial Neural Network, *International Conference on Artificial Intelligence in Engineering and Technology (ICAIET) Proceedings*, Kota Kinabalu, Sabah, 17-18 June 2002.

[8] Ahmad, A., and Abd. Hamid, M. K., 2002. Detection of Multiple Sensor Faults in a Fatty Acid Fractionation Plant Using Artificial Neural Network, *Regional Symposium on Chemical Engineering (RSCE 2002) / 16th Symposium of Malaysian Chemical Engineers (SOMChE 2002) Proceedings,* Petaling Jaya, Selangor, 28-30 October 2002.

[9] Linkens, D. and Nyongesa, Y. 1996, Learning systems in Intelligent Control: An Appraisal of Fuzzy, Neural and Genetic Algorithm Control Applications, *IEE Proc. Control Theory App.,* 134(4): 367-385.

[10] Parlos, A., Chong, K. and Atiya, A. 1994, Application of the Recurrent Multilayer Perceptron In Modelling Complex Process Dynamics, *IEEE Trans. On Neural Networks,* 2, no 2: 252-262.

[11] Draye, J., Pavisic, D. and Libert, G. 1996, Dynamic Recurrent Neural Networks: A Dynamical Analysis, *IEEE Trans. On Systems Man and Cybernetics,* Part B, vol 26, no 5: 692-706.

[12] Elman, J. 1990, Finding Structure in Time, *Cognitive Science,* 14: 1789-211.

[13] Hornik, K., Stinchcombe, M. and White, W. 1989, Multilayer Feedforward Neural Networks are Universal Approximators. *Neural Networks*, **2**, No.5, 359.

[13] HYSYS.Plant User's Manual. 2000, Hyprotech Ltd., Calgary, Canada.